Yet Another Image-processing Language

COMS W4115: Programming Languages and Translators, Fall 2010

Uday Chandrasen
Andrew Kisch
Aniket Phatak
Pranay Prabhakar

Overview & Motivation

- Why Image Processing
  - Tangible output
  - Amenable to numerical algorithms well suited for computers
  - Well understood problem domain

- Goals of YAIL:
  - Flat learning curve
  - Support for images and filters
  - Support for frequent operations
  - Ease of implementation of frequent image operations

Tutorial

- Start off – Nothing different from the ordinary:
  - Open your favourite editor.
  - Start off with the function main().
  - Write YAIL code within this function. May also create your own functions.
  - Save the file with the .YAIL extension.
  - Call the YAIL compiler on the target file.

- C based syntax
- Data types: *int, float, string, image, filter*
- Special operation: Convolution $(image \# filter)^{+}$
- Special built-in functions to aid image operations.

Tutorial

- Hello World

```
int main()
{
        int i;
        string s
        s = "Hello World";
        for ( i = 0; i < 5; i = i + 1)
        {
                print ( s );
        }
        return 0;
}
```

All variables must be declared before first "execution" line.

Built-in function

- Notice that the function declaration begins with a '{' and ends with a '}'

Tutorial

- Using the image functions

```
int main()
{
    image im;
    image im2;
    im = newImage("/home/ppp2113/yail/edwards.jpg");
    im2 = meanFilter(im);
    printImage(im2);
    printImage(edgeDetection(im));
    saveImage(im2,"/home/ppp2113/yail/edwards_edged.jpg");
    return 0;
}
```

Original

Softened

Edged

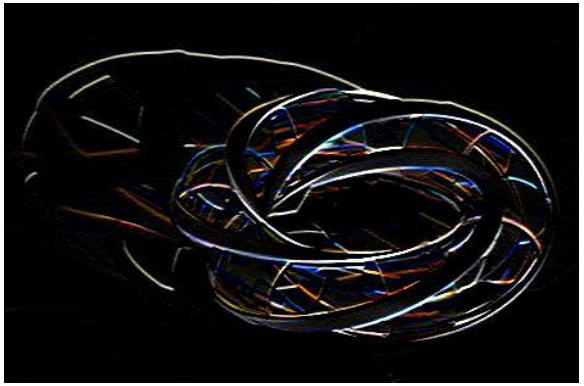
Tutorial

- Using the image filters

  Sobel's vertical derivative

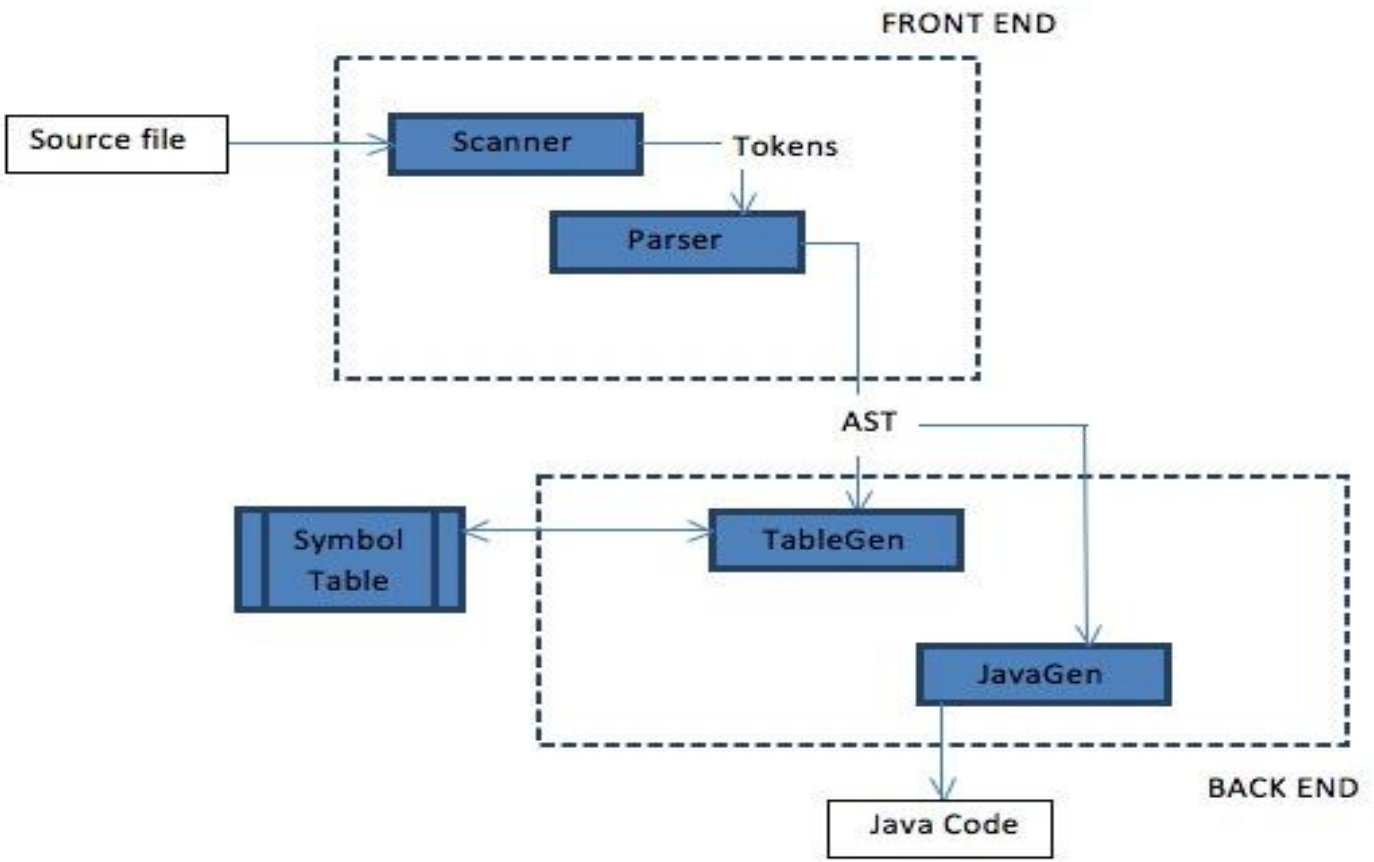
```
int main()
{
    image im;
    filter sobelY;
    im = newImage("/home/ppp2113/yail/edgyjpg");
    sobelY = { -1.0,-2.0,-1.0; 0.0, 0.0, 0.0; 1.0,2.0,1.0};
    printImage(im);
    printImage(im # sobelY );
    return 0;
}
```

Original



Edged

Implementation



Implementation Stages

- Early Stage
- Middle Stage
- Late Stage

Summary

- Goals achieved
  - Simplicity: The learning curve is believed to be reduced as the syntax is C based.
  - Images and Filters can be easily constructed and represented.
  - Rich support in terms of built-in functions
- Lessons learned
  - Adjusting to a new programming paradigm is difficult, however Ocaml is really succinct and well suited for describing grammars.
  - Appreciation of how computer languages work.
  - A big project needs discipline and ability to make hard choices to meet deadlines.
  - Cross team collaboration has a lot of difficulties. Regular communication is the key.

Thank You